



**University of
Zurich**^{UZH}

**Zurich Open Repository and
Archive**

University of Zurich
University Library
Strickhofstrasse 39
CH-8057 Zurich
www.zora.uzh.ch

Year: 2017

Bayesian graph-cut optimization for wall surfaces reconstruction in indoor environments

Michailidis, Georgios-Tsampikos ; Pajarola, Renato

Abstract: In this paper, a new method capable to extract the wall openings (windows and doors) of interior scenes from point clouds under cluttered and occluded environments is presented. For each wall surface extracted by the polyhedral model of a room, our method constructs a cell complex representation, which is used for the wall object segmentation using a graph cut method. We evaluate the results of the proposed approach on real-world 3D scans of indoor environments and demonstrate its validity.

DOI: <https://doi.org/10.1007/s00371-016-1230-3>

Posted at the Zurich Open Repository and Archive, University of Zurich

ZORA URL: <https://doi.org/10.5167/uzh-125426>

Journal Article

Accepted Version

Originally published at:

Michailidis, Georgios-Tsampikos; Pajarola, Renato (2017). Bayesian graph-cut optimization for wall surfaces reconstruction in indoor environments. *Visual Computer*, 33(10):1347-1355.

DOI: <https://doi.org/10.1007/s00371-016-1230-3>

Bayesian Graph-Cut Optimization for Wall Surfaces Reconstruction in Indoor Environments

Georgios-Tsampikos Michailidis · Renato Pajarola

Received: date / Accepted: date

Abstract In this paper, a new method capable to extract the wall openings (windows and doors) of interior scenes from point clouds under cluttered and occluded environments is presented. For each wall surface extracted by the polyhedral model of a room, our method constructs a cell complex representation, which is used for the wall object segmentation using a graph cut method. We evaluate the results of the proposed approach on real-world 3D scans of indoor environments and demonstrate its validity.

Keywords scene reconstruction · point cloud processing · LiDAR reconstruction · wall surface reconstruction · wall openings detection

1 Introduction

In recent years, an increased demand for semantically rich 3D information from building models emerged. Often this information is not available or not up-to-date from the design time, and thus the *as-is* state must be recovered.

In this context, the accurate identification of the main architectural structures of indoor environments, such as the room walls and openings, are of primary interest. Moreover, the state of the reconstruction process of building interiors is less advanced than the more well-studied case of reconstructing the exterior outlines and façades of buildings [1,2]. This is because indoor environments often contain many objects and obstacles which prevent an ideal data acquisition and easy

reconstruction of the structures of interest (see also Fig. 1). Furthermore, clutter and occlusions could also lead to erroneous model interpretations, especially in cases where parts of the wall surface and openings are absent because of an object (occluder) in front of them. In addition, windows and other highly reflective surfaces can drastically affect the acquired models by introducing large-scale artifacts and causing missing data, due to their unfavorable material properties. Windows in particular, which are mainly shapeless, textureless and transparent, constitute an important source of reconstruction errors and artifacts. Moreover, doors are hard to be detected, since their reconstruction depends strongly on the wall geometry and the descriptive features of the door.

Recently a number of methods have been proposed to overcome these limitations and reconstruct the structural elements of indoor scenes. Some of these rely on *a priory* knowledge obtained from training models and supervised learning algorithms in order to provide evidences for structural wall elements [2–4], while others combine indoor and outdoor scans in order to avoid heavy indoor occlusions [5], or they do not consider occlusion at all [6]. Furthermore, many techniques combine both 3D point and color image data from cameras [1] or stereo (depth) images [7] in order to classify the indoor objects and recover the architecture of the scene, while there are methods which try to keep in a minimum level the clutter and occlusion, scanning only indoor areas that have only essential interior objects and lack of commonly used furnitures [6].

Hence the purpose of this paper is to propose a new method capable of recovering automatically the architectural components of wall features in indoor environments, i.e. windows and doors, under the presence of significant amounts of clutter and occlusion, without

G.-T. Michailidis and R. Pajarola
University of Zurich, Dept. of Informatics, Binzmühlestr. 14,
8050 Zurich, Switzerland
E-mail: {gtmichail, pajarola}@ifi.uzh.ch



Fig. 1: Examples from heavy occlusions and clutter that can occur in indoor environments. Occluders are also visible, whose removal will cause also occlusions in wall surfaces of interest, affecting strongly the wall openings reconstruction.

requiring any tuning or using any additional imagery or depth data, and avoiding the restrictions or limitations introduced by the aforementioned techniques. Our method uses as input a set of 3D point cloud scans, and it assumes that the room’s interior is composed by vertical walls, which holds true for the vast majority of scenarios.

We demonstrate the validity of our approach by presenting results from both window and door reconstructions on real-world 3D scans of typical indoor environments. In our experiments, we evaluated our method on adverse datasets and wall surfaces with occlusions of up to 65% and observed wall surface area of less than 35% (see Section 7), while the majority of previous methods would rather perform poorly at these occlusion levels.

2 Related Work

In recent years, several approaches have been proposed for the automatic generation of indoor models. Most of them focus on the successful reconstruction of architectural structures, such as the room walls, floors and ceilings, but only few of them aim to also include the structural wall openings such as windows and doors [2].

A comprehensive review about the building reconstruction methods can be found in [8,9]. Since our proposed method mainly focuses on the detailed modeling of wall surfaces in real-world environments, we will primarily review the surface-based methods which are closely related to ours and outline a detailed pipeline process for estimating the structural wall elements.

Significant improvements towards automation in architectural room elements reconstruction have been made by Xuehan et al. [4], where raw 3D point cloud data is automatically converted into a compact and semantically rich information model. Authors in [10] present a

graph-cut based reconstruction technique in order to reconstruct the wall surfaces from indoor scenes, without however identifying their openings. In [1] the authors combine information from 3D point clouds and images in order to reconstruct and classify the indoor scenes, without being able however to identify the windows on the walls.

For identification of windows in indoor environments, several methods have been developed, either relying on analyzing data density and classifying low-density areas as wall openings ([11,12]), or by calculating the intersections of merged planes, followed by semantically labeling them using machine learning techniques [7]. Alternative techniques based on spatial and functional relationships between nearby structures and spaces are presented in [2] and [4], which are also able to identify and model the main structural components of room interiors, despite the presence of clutter and occlusion.

Recent methods on fully-automatic reconstruction of indoor scenes are proposed also in [6,13–17]. However, these techniques use specialized equipment [13, 15], which limits their application field and usage, or depend on the Manhattan-world assumption which is often too restrictive for indoor scenes [6,14]. Furthermore, fully automated 3D room modeling from interior scans has been achieved only under ideal conditions and lack of occlusions [6], and the capturing and incorporation of reliable as-is data remains very challenging, error-prone and time-consuming, considering the strict constraints in accuracy, time and cost [13,16].

More recently, a number of papers try to solve the indoor reconstruction problem by combining data from laser scanners and cameras in order to avoid the above mentioned restrictions. Previtali et al. [5] presented a dual technique for modeling indoor and outdoor environments, creating semantically-enriched building models under occlusions and clutter. Similarly, Zhang et al. [3] identify automatically window regions on exterior facing façades of buildings. By exploiting information from visual imaging modality, they can identify also the indoor window regions. However, such methods assume large regular grid of patterns for windows detection, an assumption which does not necessarily hold for indoor views of rooms, or use training models to boost performance, a time-consuming process with many limitations, such as the limited amount of training data or the manual intervention in the training process.

3 Overview of Method

We formulate the wall features extraction as a graph-cut optimization problem on a 2D cell complex defined by the line features of the reconstructed wall surface. A

general diagram presenting the pipeline process of the proposed method is shown in Fig. 2.

First, the scene of interest is scanned by a LiDAR scanner from one or more positions in order to acquire the input scans. After scan registration and positioning, all input scans are merged together to output a non-structured 3D point cloud, which forms the input dataset of our pipeline. Additionally, the main architectural elements of the scene are identified and extracted, using a wall surface segmentation technique which relies on an occlusion-aware process extracting planar patches as candidate wall segments as described in [17].

Once the planar regions constituting the wall surfaces are extracted, each wall segment is processed separately by an α -shapes algorithm [18] in order to compute the outline shape of its point set. The computed outline polygons constitute a higher order representation of the wall surface features.

Next, a robust RANSAC [19] based line model fitting algorithm is applied to the extracted outline points of the wall surface. The identified lines are then further clustered to get a smaller number of good representative model lines, which will form the basis of a 2D cell complex. The wall feature cell complex is built from all intersections of the representative feature lines from each cluster, and its cells are used to define a dual adjacency graph. The 2D cell decomposition is extended to the borders of the associated wall surface, which are defined by boundary edges.

In parallel, a voxel representation of the scene is generated, and the part of the scene that lies between the laser scanner position and the selected wall surface is obtained for occlusion detection. In this process, the occluders and the occluded points are identified and labeled, in order to create an enhanced wall surface representation.

On the wall's 2D cell complex a graph-cut optimization problem is defined to solve a max-flow/min-cut problem that eventually identifies the window and door features. The graph-cut optimization segments the cells into regions belonging to the solid wall and into areas representing the open parts, i.e. windows and doors.

4 Feature Outlines

4.1 Alpha-Shapes Outline & Refinement

From the registered input 3D point cloud, wall planes are detected and extracted for further processing. Each wall consists of a set of points $\mathcal{P} = \{\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{N-1}\}$ and is defined by its plane normal vector \mathbf{n} and some point $\mathbf{p}_i \in \mathcal{P}$, the wall's oriented bounding box (OBB),

as well as the wall outline polygon from the intersection of OBB with the wall plane (see also Fig. 3).

In the 2D projection of the wall plane points \mathcal{P} , their α -shape boundary outline is first computed and refined, in order to eliminate the introduced bias at borders and improve the computed raw outline [18], see also Fig. 4(b). The user specified parameter α (radius) determines the complexity and smoothness of this boundary and can be chosen conservatively, such as to be sufficiently large compared to the inverse of the scan point density.

As a further improvement, the extracted polyline α -shapes that do not fulfill predefined criteria are filtered out. More specifically, the minimum polygon surface area was set to 15cm^2 , assuming that the room windows or doors are bigger than this size, while the open polylines and isolated α -extreme points (i.e. outliers) were also filtered out, assuming that they cannot form the desired wall feature shapes. One such an example is presented in Fig. 4.

4.2 Line Model Fitting and Clustering

The α -shapes boundaries described above only produce a rough estimation of the true underlying object boundaries as shown in Fig. 4(c). In order to get regularized boundaries, a RANSAC algorithm was used to robustly fit straight line segments to the refined α -shapes boundary. Let \mathcal{A} be the α -shape point set. Then, the best l line models $\mathcal{Q} = \{\mathcal{Q}^0, \mathcal{Q}^1, \dots, \mathcal{Q}^{l-1}\}$ for \mathcal{A} are estimated (see also Fig. 5(a)).

Moreover, instead of keeping all estimated line models, the line space is further reduced to a set of mode (i.e. hypothesis, representative) line models $\mathcal{M} = \{\mathcal{M}^0, \mathcal{M}^1, \dots, \mathcal{M}^{m-1}\}$ using the mean-shift clustering [20], and each line model in \mathcal{Q} is assigned to one of the modes in \mathcal{M} . The clustering is first performed only on the orientations, extracting the major line directions, and then for each representative orientation a 1D mean-shift clustering finds the most likely line offsets, similar to [14] (see also Fig. 5(b)). For the line clustering, we furthermore assume that the wall's boundary lines are almost parallel to the corresponding polygonal line segments of windows and door.

The resulting set of line clusters corresponds to a partitioning of the line models \mathcal{Q} and each cluster defines a representative mode line from \mathcal{M} .

4.3 Cell Complex Creation

Given the clustered mode lines, a 2D cell complex representation \mathcal{I} of the underlying wall surface is built.

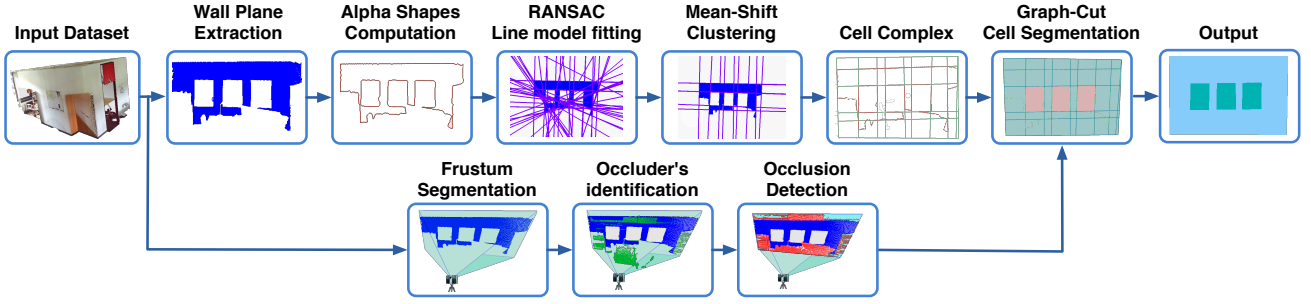


Fig. 2: Block diagram of the proposed reconstruction technique.

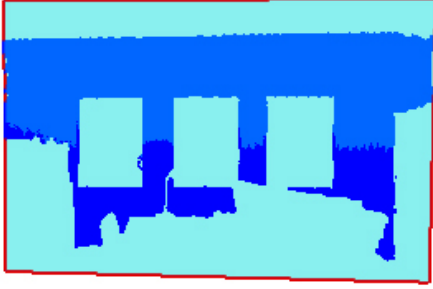
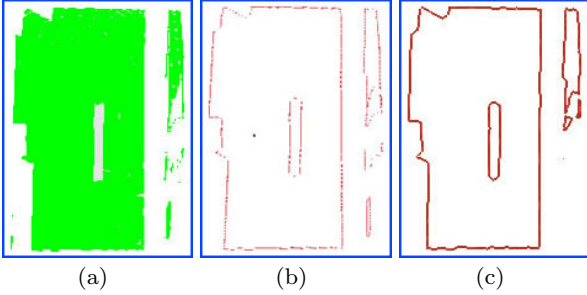


Fig. 3: Extracted wall plane. Interior wall surface points, indicated in blue, and the wall outline polygon, indicated in red.

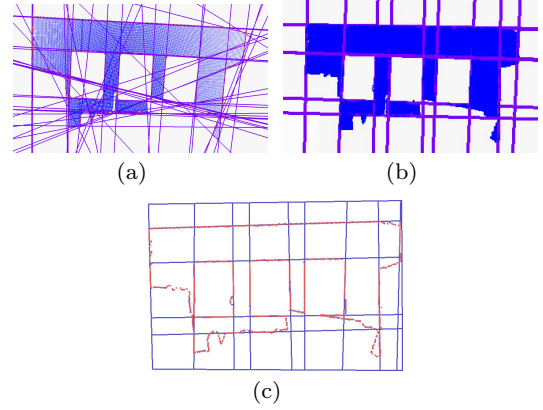
Fig. 4: (a) Wall surface (green) points representing a door. (b) α -shape boundary points. (c) Refined α -shape points and the corresponding rough outline shapes.

This cell complex is defined by all intersections of the representative lines in the wall plane and bounded by the wall's outline polygon as shown in Fig. 5.

Note here also that the cells in the cell complex are explicitly associated with the α -extreme points from \mathcal{A} that lie on the line segments of the representative mode lines which form their boundaries, since they are used as a contribution factor to the estimation of the weights in our graph-cut implementation.

5 Occlusion Detection

Once all mode lines are detected, the occluded areas and the actual empty space in the wall surface have to be determined. This step is important, since the various

Fig. 5: Cell complex construction. (a) Detected wall surface and the line models fitted to the (red) α -shapes outline. (b) Representative mode lines after mean-shift clustering. (c) 2D cell complex \mathcal{I} (blue) imposed by the representative mode lines on the wall outline polygon.

objects inside rooms introduce occlusions and clutter, which significantly affect the outcome of the wall surface reconstruction producing holes which have to be distinguished from real wall openings.

Thus similar to [2], we conduct an occupancy analysis on the wall surface in order to create a voxel representation of space and identify the clutter and non-clutter areas. In our approach, the sample points in a voxel are approximated by their centroid instead of the voxel center for accuracy reasons.

The result of this voxelization process is an oriented uniform grid of cells over the wall surface. The grid cells are then classified according to their occupancy state into three categories *empty*, *occluded* and *occupied*. The results from this occlusion analysis process are illustrated in Fig. 6.

The information derived from the occlusion detection process constitutes an additional decision making factor in the wall segmentation process, as it is described in the next section.

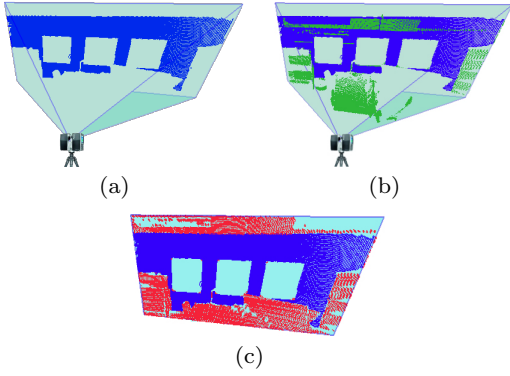


Fig. 6: Occlusion detection. (a) Wall surface (blue), (b) detected occluders (green), and (c) the recovered occluded areas (red).

6 Graph Cut Computation

6.1 Problem Formulation

Given the 2D cell complex, a weighted undirected graph G is defined by associating each cell c_p within the cell complex structure \mathcal{I} to a vertex $v \in V$ in G and tagging each vertex by a semantic label from the label set $\mathcal{L} = \{\mathcal{L}_{emp}, \mathcal{L}_{occ}\}$ for empty or occupied cells, respectively. Apart from the set of nodes (or vertices) V , the graph contains also a set E of edges $e(u, v)$ with non-negative weights (capacities), which connect adjacent nodes and are called n-links. The node set includes also two additional terminal nodes, the source s and the sink t , and the corresponding edges connecting the nodes with the two terminals are called t-links.

On this enhanced dual graph of the cell complex, an optimal labeling can be found by minimizing an objective (cost) function applying a minimum graph-cut approach to Markov Random Fields (MRFs) [21, 22]:

$$E(f) = \sum_{p \in V} \mathcal{D}_p(f_p) + \sum_{\{p, q\} \in N_G} V_{p, q}(f_p, f_q) \cdot T(f_p \neq f_q), \quad (1)$$

where $\mathcal{F} = \{f_0, \dots, f_{n-1}\}$ is a binary vector of assigned labels $f_j \in \mathcal{L}$ defining the segmentation. $\mathcal{D}_p(f_p)$ is the unary data term, $V_{p, q}(f_p, f_q)$ is the submodular smoothness penalty term [23], N_G is the neighborhood system and the comparison predicate $T(\cdot)$ is 1 if its argument is true and 0 otherwise. Additionally, the two energy terms in Eq. 1 represent the two types of edges in the graph, i.e. the t-links and n-links, while in our case the neighboring system was set to a 4-connected lattice of cells.

In our framework, the unary data term $\mathcal{D}_p(f_p)$ is defined using a Bayesian probabilistic model. Specifically,

we represent the wall cell complex \mathcal{I} by a vector of n elements, $\mathbf{z} = [z_0(f_0), \dots, z_{n-1}(f_{n-1})]$, indicating the estimated class probabilities $z_p(f_p) = Pr(f_p | c_p)$ of a cell c_p to belong to label f_p . Applying Bayes' rule, this probability can be expressed as

$$Pr(f_p | c_p) = \frac{Pr(f_p)Pr(c_p | f_p)}{Pr(c_p)} \propto Pr(f_p)Pr(c_p | f_p), \quad (2)$$

where $Pr(c_p)$ is a normalizing constant which is independent from f_p and can be omitted, while $Pr(f_p)$ and $Pr(c_p | f_p)$ are computed using spatial information from the cell complex. The prior probability distribution $Pr(f_p)$ of the state f_p encodes our belief in the likelihood of various cell labels, while $Pr(c_p | f_p)$ is the conditional probability of seeing the cell c_p given the particular cell label f_p . In our case, it is based on the cell density function

$$\varrho_p = \gamma_D \frac{|Cov_{c_p}|}{\mathcal{S}_{c_p}} = \gamma_D \frac{|\mathcal{P}_{c_p} - \mathcal{A}_{c_p}|}{\mathcal{S}_{c_p}}, \quad (3)$$

where Cov_{c_p} is equal to the number of points covering the cell c_p (\mathcal{P}_{c_p}) minus the number of α -extreme points (\mathcal{A}_{c_p}) that lie in the cell's boundary line segments. \mathcal{S}_{c_p} is the cell's area and γ_D a scaling factor (see also Table 1).

The conditional probability $Pr(c_p | f_p)$ of an arbitrary cell c_p is then computed by

$$Pr(c_p | f_p) = \frac{\varrho_p}{K} e^{-|\mathcal{A}_{c_p}|/Cov_{c_p}}, \quad (4)$$

where K is a normalization factor. Also, given the condition f_p and presuming the equivalent cell state (i.e. occupied or empty), \mathcal{P}_{c_p} is included (or not) in the computation of cell's density function in Eq. 3, assuming that in the case of an empty cell, any (internal) point other than the boundary points should be an outlier.

Inspired by [24], the prior probability $Pr(f_p)$ is estimated based on a distance function, which relies on specific features of cells in \mathcal{I} . Let $\mathcal{B} = \{b_{occ}, b_{emp}\}$ be a set of two labeled cells from \mathcal{I} , called *base* cells, which are actually the cells for which we can have a high certainty for their labels. Let \mathbf{x}_{b_m} denote the feature vector describing the base cell $m \in \{occ, emp\}$ and let \mathbf{x}_{c_p} denote the feature vector of an arbitrary cell c_p . Let also $\mathbf{y}_{b_m c_p}$ denote the vector containing the L_2 norms between the individual features describing b_m and c_p , i.e. $\mathbf{y}_{b_m c_p}[u] = \|\mathbf{x}_{b_m}[u] - \mathbf{x}_{c_p}[u]\|$.

The selection of features is critical for the definition of feature vectors and base cells, and they should be sufficiently simple for fast computation. In our case, it has been shown that the density ϱ_p as a simple feature is sufficient in order to define the certainty of a cell being a base cell. However, other features such as the

ratio $|\mathcal{A}_{c_p}|/Cov_{c_p}$ and the coverage Cov_{c_p} gave similar results.

The base cells selection was thus based on ϱ_p , assuming that the cell with the highest density should belong to the wall surface (label \mathcal{L}_{occ}), and equivalently, the cell with the minimum density should belong to a wall opening (label \mathcal{L}_{emp}).

We then estimate which cells are likely to have the same label as the associated base cell, using the distance function

$$d(b_m, c_p) = \beta_\epsilon \mathbf{y}_{b_m c_p} \quad \text{with} \quad \beta_\epsilon = \frac{1}{\gamma_r \varrho_{b_m}}, \quad (5)$$

where the weight β_ϵ defines the range of cell density in which the cells are likely to have the same label, and γ_r is a coefficient defining the deviation of cell's density from the base cell density ϱ_{b_m} . In our case, this was fixed to 0.2, meaning that every cell whose density is less than 20% from ϱ_{b_m} will be assumed to belong to the same set as the base cell b_m .

To determine the prior probability $Pr(f_p)$, we first set a decision boundary using the inequality $d(b_m, c_p) \leq 1$, where a value of 0 corresponds to perfect similarity. It has been shown in [24] that such a threshold gives empirically good results. Hence the prior probability $Pr(f_p)$ is given by:

$$Pr(f_p) = n^{-1} |\{c_p \mid d(b_m, c_p) \leq 1\}|, \text{ when } f_p \equiv \mathcal{L}_m \quad (6)$$

6.2 Wall Segmentation

From $Pr(c_p \mid f_p)$ and $Pr(f_p)$ we can estimate the class probabilities $Pr(f_p \mid c_p)$ (i.e. $z_p(f_p)$), which are used for defining the unary data term $\mathcal{D}_p(f_p)$. But before setting the edge weights, we also impose certain regional hard constraints for segmentation, by indicating certain cells that should belong to wall surface or wall openings [22]. Generally, a regional constraint on G is a vertex in V which is adjacent to a terminal node and is called *seed* node.

Based on two empirical constants κ_H and κ_L set to 0.9 and 0.2, respectively, we define the initial seed nodes as in Eq. 7. Note that this initialization also determines the connection to the associated terminal nodes with a sufficiently large (theoretically infinite) weight.

$$f_p = \begin{cases} \mathcal{L}_{occ} & , \text{ if } z_p(\mathcal{L}_{occ}) \geq \kappa_H \\ \mathcal{L}_{emp} & , \text{ if } z_p(\mathcal{L}_{emp}) \leq \kappa_L \end{cases} \quad (7)$$

Next, the edge weights of t-links connecting non-seed vertices to the terminal nodes are computed. For the source terminal link, the edge weight is given by

$$w_p^{(s)} = \beta_w \frac{z_p - \kappa_L}{\kappa_H - \kappa_L} \quad (8)$$

where β_w is a sufficiently large constant which ensures a feasible flow in the graph. Similarly, for the sink terminal link, the edge weight is defined as

$$w_p^{(t)} = \beta_w \frac{\kappa_H - z_p}{\kappa_H - \kappa_L} \quad (9)$$

From Eq. 8 and Eq. 9 we can assign the t-links weights to the unary data term $\mathcal{D}_p(f_p)$ as [22]:

$$\mathcal{D}_p(f_p) = w_p^{(s)} + w_p^{(t)} \quad (10)$$

Furthermore, we define the edge weights used for the n-links as

$$w_{p,q} = \begin{cases} \kappa_n (\mathcal{R}_{dens} + \mathcal{R}_a) & , \text{ if } p \neq q \\ 0 & , \text{ if } p = q \end{cases} \quad (11)$$

where \mathcal{R}_{dens} and \mathcal{R}_a are the contributions of certain cell features, and κ_n is a properly selected constant, which ensures that the n-link weights will be in the same range as the non-seed t-links.

The term \mathcal{R}_{dens} is used to measure the difference of cell densities and is defined as:

$$\mathcal{R}_{dens_{p,q}} = e^{-\frac{(\varrho_p - \varrho_q)^2}{2\sigma^2}} \quad (12)$$

The term \mathcal{R}_a computes the difference of α -extreme points, which is an indicator if a cell is near to a transition region between the wall surface and a wall opening:

$$\mathcal{R}_{a_{p,q}} = |\mathcal{A}_{c_p}|^2 - |\mathcal{A}_{c_q}|^2 \quad (13)$$

After assigning the weights to the neighboring node n-links, the problem can be mapped to a graph-cut problem with its min-cut cost $|C|$ equalling $E(f)$, which is constructed using the Eq. 10 and the following edge weights assignments [22]:

$$V_{p,q}(f_p, f_q) = w_{p,q} \quad (14)$$

Then, the exact solution of the global minimum cut with only two terminal nodes can be computed from Eq. 1 in polynomial time [22]:

$$E(f) = \sum_{p \in V} (w_p^{(s)} + w_p^{(t)}) + \sum_{\{p,q\} \in N_G} w_{p,q} \cdot T_{p \neq q}, \quad (15)$$

Table 1: List of parameters used in graph-cut segmentation.

Parameter	Value	Parameter	Value
γ_D	10^{-3}	κ_H, κ_L	0.9, 0.2
K	autom. computed	β_w	10^4
γ_r	0.2	κ_n	autom. computed

7 Experimental Results and Discussion

In this section, we present the quantitative and qualitative results from the evaluation of our method. We have tested it on many real-world datasets, which present different wall surfaces and occlusion levels. The selected walls include windows and doors, and our algorithm is evaluated in both cases without manual tuning, as opposed to other techniques in the literature which are particularly tuned to identify only one of them (e.g. such as [1,3]). The datasets were captured by a LiDAR range scanner and were processed on an Intel Core i7 2.8 GHz CPU computer, without any multithreading technique.

Tbl. 2 and 3 list the relevant statistics for the used datasets. Tbl. 2 shows the properties of the datasets in the first three columns and the experimental results after each main stage of the pipeline process. It also includes the ground truth of the wall openings based on human observations, as well as the number of extracted wall openings from our algorithm. Overall we can see that there is only one misclassification error in all tests, in OFFICE5, while all other wall openings were correctly detected and segmented. Additional information about our experiments is also given in Tbl. 3, where the corresponding timings are presented.

The difference observed in processing times for the α -shapes (AS) in Tbl. 3 between OFFICE3 and the other environments is related to the smaller number of points in the extracted wall plane (see also Tbl. 2), but also to the reduced number of Delaunay triangulations necessary for the outline wall polygon computation. It is also evident that the graph-cut computation is the most costly part, and that the challenging occlusions present in some datasets lead to an increased computation time.

Fig. 7 shows the results of the main pipeline stages of our method for the OFFICE1 dataset. In Fig. 7(a), the room interior is presented for convenience with the walls in different colors. Fig. 9 shows the results of wall surface segmentation from three more office environments. The OFFICE2 dataset (Fig. 9(a)) contains an open doorway and a wardrobe in front of the wall, while there are many other objects that create additional occlusions. From the extracted results we can note that although the wardrobe has hidden a big rectangular part on the right side of the extracted wall surface, which could be wrongly interpreted as a doorway too, our method was able to successfully classify it as part of the wall.

The OFFICE3 dataset (Fig. 9(b)) contains a big window with three adjacent frames, which covers the most part of the wall, but also contains many occluders, which hide a large part of the remaining wall sur-

face and parts of the window frames too. Nevertheless, our algorithm was able to correctly identify the wall openings. The OFFICE4 dataset (Fig. 9(c)) contains a closed door along with some closets in front of the wall, which cover part of it but also part from the door frame. Despite the challenging room environment, the doorway was accurately detected after the graph-cut segmentation.

Fig. 8 shows the most challenging dataset of our experiments. It contains a complex and irregular combination of window frames, which is an exceptionally difficult setting for the laser scanner, resulting in a highly cluttered and occluded wall surface (up to 65% occluded), with many corrupted window frame segments. Despite these challenges, our algorithm detected correctly almost all window frames, although some cells from the cell complex were erroneously interpreted due to its complexity.

8 Conclusions

We have presented a method capable to reconstruct automatically the wall surfaces of the interior of complex indoor environments. Our wall features segmentation pipeline was tested in real-world scenarios and performed adequately well under clutter and occlusion.

Although our approach considers only buildings that have planar and vertical walls, it can detect efficiently both windows and doors, as opposed to the majority of the current methods in literature.

References

1. L. Díaz-Vilariño, J. Martínez-Sánchez, S. Lagüela, J. Armesto, and K. Khoshelham. Door recognition in cluttered building interiors using imagery and lidar data. In *Proceedings International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XL-5, pages 203–209, June 2014.
2. Antonio Adan and Daniel Huber. 3D reconstruction of interior wall surfaces under occlusion and clutter. *3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 275–281, May 2011.
3. Richard Zhang and Avideh Zakhori. Automatic identification of window regions on indoor point clouds using LiDAR and cameras. *IEEE Applications of Computer Vision*, pages 107–114, March 2014.
4. Xiong Xuehan, Adan Antonio, Akinci Burcu, and Huber Daniel. Automatic creation of semantically rich 3D building models from laser scanner data. *Automation in Construction*, 31:325–337, January 2013.
5. Mattia Previtali, Marco Scaioni, Luigi Barazzetti, and Raffaella Brumana. A flexible methodology for outdoor/indoor building reconstruction from occluded point clouds. *Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(3):119–126, September 2014.

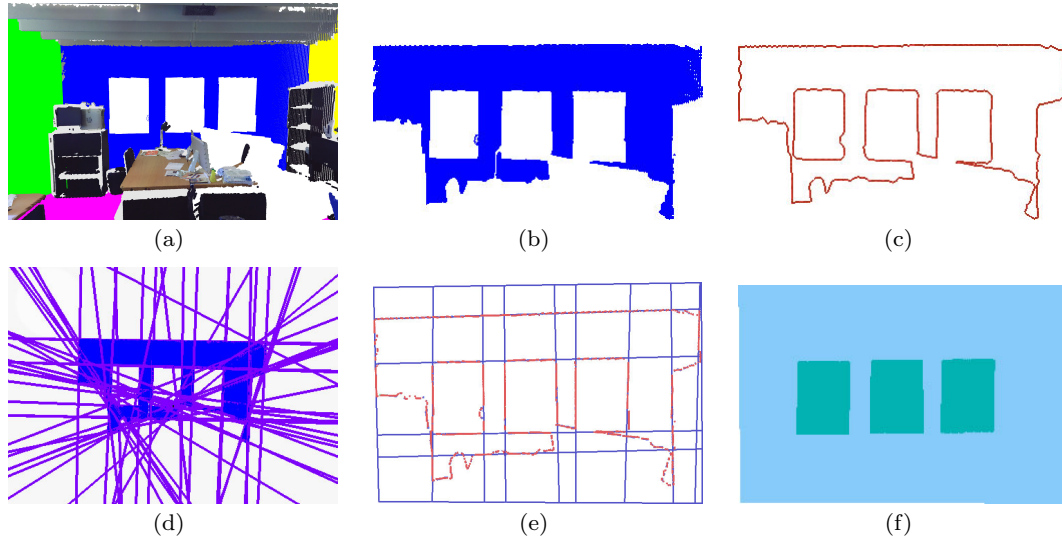


Fig. 7: Results from the main pipeline stages for OFFICE1. (a) The room environment with the identified wall surfaces (colored), (b) the selected wall surface, (c) the Alpha Shapes, (d) the line models, (e) the cell complex and (f) the graph-cut result.

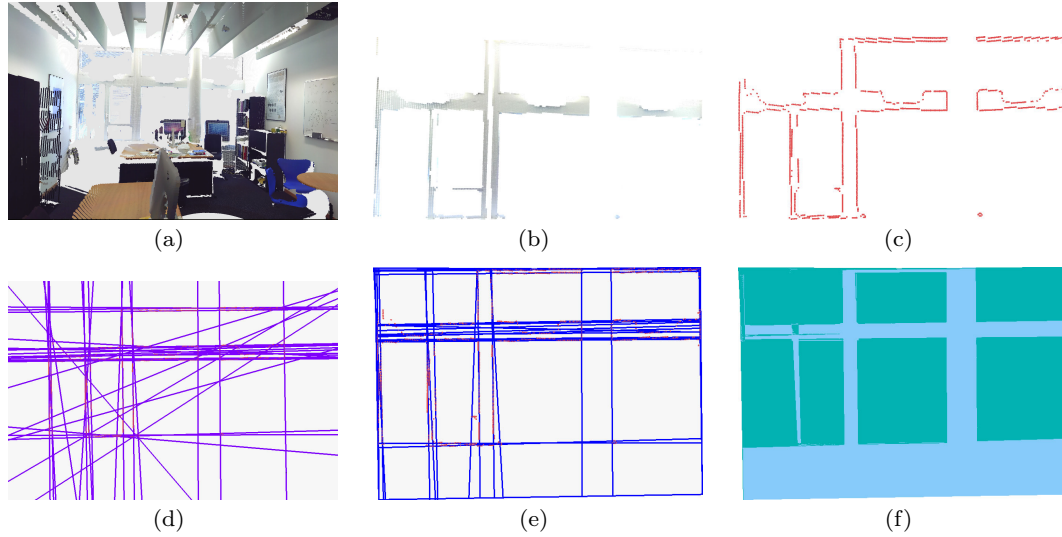


Fig. 8: Results from the main pipeline stages for OFFICE5. (a) The room environment, (b) the selected wall surface, (c) the Alpha Shapes, (d) the line models, (e) the cell complex and (f) the graph-cut result.

Table 2: Datasets description and statistics, showing the number of: points in the extracted wall plane ($\#Pnts/Wall$), line models detected by the line model fitting algorithm ($\#LM$), lines which comprise the cell complex ($\#CC\ lines$), occluded points and their percentage in wall surface ($\#OccPnts(\frac{\#OccPnts}{\#OccPnts+\#Pnts/Wall} \%)$), Alpha Shapes points before ($\#AS$) and after ($\#AS(ref.)$) the refinement process, cells in cell complex ($\#Cells$), wall openings in ground truth and detected by our method ($\#W_Open.(GT/det.)$).

Dataset	#Points	#Scans	#Pnts/Wall	#LM	#CC lines	#OccPnts (%)	#AS	#AS(ref.)	#Cells	#W_Open.(GT/det.)
OFFICE1	626058	1	24629	49	18	10071 (29%)	1367	1365	60	3/3
OFFICE2	2685833	1	131704	25	12	6998 (5%)	2311	2037	25	1/1
OFFICE3	2685833	1	16456	25	16	12214 (43%)	2123	2098	45	3/3
OFFICE4	8139527	3	256817	8	12	48380 (16%)	1789	1131	25	1/1
OFFICE5	626807	1	3747	38	32	6924 (65%)	1149	951	224	8/7

6. Angela Budroni and Jan Böhm. Automatic 3D modelling of indoor manhattan-world scenes from laser data. *International Archives of the Photogrammetry, Remote Sens-*

ing and Spatial Information Sciences, XXXVIII:115–120, February 2010.

Table 3: Processing times for our datasets, for each processing stage of the pipeline process, i.e. for Alpha Shapes computation (AS), line model fitting (LM), occlusion detection (OC), cell complex construction and graph-cut computation (CC+GC).

Dataset	AS (sec)	LM (sec)	OC (sec)	CC+GC (sec)	Total (sec)
OFFICE1	1.44	0.15	0.37	17.81	19.76
OFFICE2	7.54	0.04	1.36	19.78	28.72
OFFICE3	0.85	0.15	1.41	8.12	10.71
OFFICE4	7.39	0.01	4.31	14.12	25.82
OFFICE5	0.18	0.09	1.09	14.17	15.53

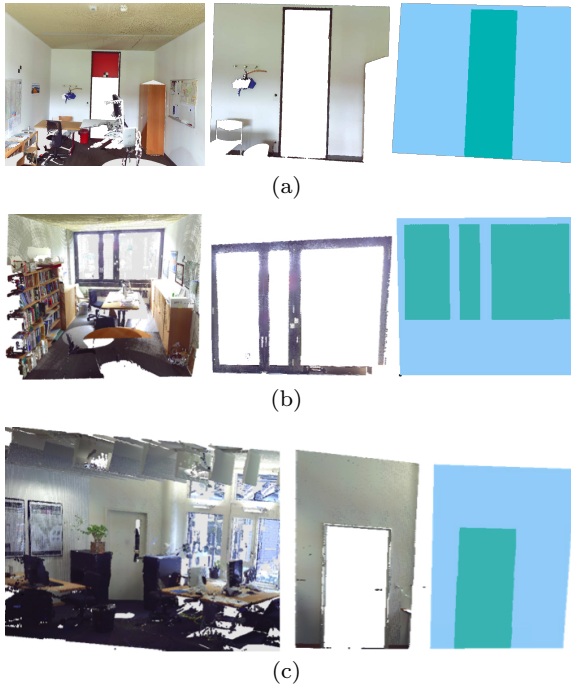


Fig. 9: Wall surface reconstruction for (a) OFFICE2, (b) OFFICE3 and (c) OFFICE4. For each dataset, the input point cloud (left), the reconstructed wall plane (middle) and the wall openings segmentation (right) are presented.

7. Remus-Claudiu Dumitru, Dorit Borrmann, and Andreas Nüchter. Interior reconstruction using the 3D Hough Transform. In *Proceedings International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XL-5/W1, pages 65–72, February 2013.
8. Pingbo Tang, Daniel Huber, Burcu Akinci, Robert Lipman, and Alan Lytle. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in Construction*, 19(7):829–843, June 2010.
9. Rebekka Volk, Julian Stengel, and Frank Schultmann. Building information modeling (BIM) for existing buildings - Literature review and future needs. *Automation in Construction*, 38:109–127, March 2014.
10. Sven Oesau, Florent Lafarge, and Pierre Alliez. Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. *ISPRS Journal of Photogrammetry and Remote Sensing*, 90:68–82, 2014.
11. Alexander Hinneburg and Daniel A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *Proceedings International Conference on*

- Knowledge Discovery and Data Mining*, pages 58–65, 1998.
12. Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.
13. Arayici Yusuf. An approach for real world data modelling with the 3D terrestrial laser scanner for built environment. *Automation in Construction*, 16(6):816–829, 2007.
14. Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Manhattan-World stereo. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 1422–1429, June 2009.
15. Brian Okorn, Xuehan Xiong, Burcu Akinci, and Daniel Huber. Toward automated modeling of floor plans. In *Proceedings Symposium on 3D Data Processing, Visualization, and Transmission*, volume 2, pages 17–20, May 2010.
16. Daniel Huber, Burcu Akinci, Antonio Adan Oliver, Engin Anil, Brian E. Okorn, and Xuehan Xiong. Methods for automatically modeling and representing as-built building information models. In *Proceedings NSF CMMI Research Innovation Conference*, January 2011.
17. Claudio Mura, Oliver Mattausch, Alberto Jaspe Villanueva, Enrico Gobbetti, and Renato Pajarola. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. *Computers and Graphics*, 44:20–32, 2014.
18. Herbert Edelsbrunner and Ernst P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, January 1994.
19. Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226, June 2007.
20. Dorin Comaniciu and Peter Meer. Mean Shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.
21. Yuri Boykov, Olga Veksler, and Ramin Zabih. Markov Random Fields with efficient approximations. In *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 648–655, June 1998.
22. Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient N-D image segmentation. *International Journal of Computer Vision*, 70(2):109–131, November 2006.
23. Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Graph cut based image segmentation with connectivity priors. *IEEE Computer Vision and Pattern Recognition*, pages 1–8, June 2008.
24. Andrea Frome, Yoram Mali, and Jitendra Malik. Image retrieval and recognition using local distance functions. In *Proceedings Advances in Neural Information Processing Systems 19*, volume 19, pages 417–424, October 2006.